CFD Analysis using Multigrid Algorithm

A project report submitted in partial fulfillment of the requirements

for the degree of

Bachelor of Technology (Mechanical Engineering)

By

Sunil Kumar Rath (10503069)
Session 2008-09

Under the guidance of

Prof. S.K. Mahapatra

National Institute of Technology Rourkela

Rourkela-769008

Orissa

National Institute of Technology Rourkela
Rourkela-769008, Orissa

CERTIFICATE

This is to certify that the Project entitled **"CFD Analysis using Multigrid Algorithm"** submitted by **Sunil Kumar Rath** in partial fulfillment of the requirements for the award of **Bachelor of Technology Degree** in **Mechanical Engineering** Session 2005-2009 at **National Institute of Technology, Rourkela** is an authentic work carried out by him under my supervision and guidance.

Place: NIT Rourkela
Date:

(Prof. S.K.Mahapatra)
Department of Mechanical Engineering
NIT Rourkela

ACKNOWLEDGEMENT

I would like to express my gratitude towards all the people who have contributed their precious time and effort to help me, without whom it would not have been possible for me to understand and complete the project.

I would like to thank Prof S.K. Mahapatra, Department of Mechanical Engineering, my Project Supervisor for his guidance, support, motivation and encouragement throughout the period this work was carried out. His readiness for consultation at all times, his educative comments, his concern and assistance even with practical things have been invaluable.

I am grateful to Prof R.K.Sahoo, Head of Department, Mechanical Engineering for providing the necessary facilities in the department.

Date: 12/05/09 SUNIL KUMAR RATH
(10503069)

Abstract

The multigrid algorithm is an extremely efficient method of approximating the solution to a given problem. The functions involved in the calculations are all discrete, or discontinuous, and are represented by an array of values taken from equally-spaced points along their range - a "grid." The algorithm's efficiency lies in the fact that once an approximate solution to the problem is found its accuracy can be improved using calculations on increasingly sparse grids which require less processing power.

In this project the theory behind the multigrid algorithm was studied and a computer program was written which demonstrates the use of this algorithm in solving the problem of natural convection. Stream function-Vorticity approach and the Bossinesq approximation were used in the programs. Also, the same problem was solved using the Multigrid algorithm using Fluent software. The results obtained were matched with the analytical results.

Keywords: multigrid algorithm, natural convection

List of Figures

CONTENTS

Chapter 1

Literature Review

Computational fluid dynamics has been widely applied in areas such as aerospace, automobile and materials manufacturing industries. The applications include power production processes, heating and air conditioning of buildings, design of electronic circuits, prediction of environmental pollution, simulation of blood flow through the human body and design of artificial human limbs. Since the processes under consideration have such an overwhelming impact on human life, we should be able to deal with them effectively.

The goal of CFD is to provide an understanding of the nature of these processes and to help in designing new processes. However, the real world processes are usually too large and too complicated to simulate due to the computing and memory limits. Recent developments in multi grid algorithms which accelerate the convergence of solution process and advances in parallel computers offer the promise of providing orders of magnitude increase in computational power.

The multigrid algorithm is an extremely efficient method of approximating the solution to a given problem. The functions involved in the calculations are all discrete, or discontinuous, and are represented by an array of values taken from equally-spaced points along their range - a "grid." The algorithm's efficiency lies in the fact that once an approximate solution to the problem is found its accuracy can be improved using calculations on increasingly sparse grids which require less processing power.

Already in the sixties R.P. Fedorenko developed the first multigrid scheme for the solution of the Poisson equation in a unit square. Since then, other mathematicians extended Fedorenko's idea to general elliptic boundary value problems with variable coefficients.
However, the full efficiency of the multigrid approach was realized after the works of A. Brandt and W. Hackbusch . These authors also introduced multigrid methods for nonlinear problems like the multigrid *full approximation storage* (FAS) scheme.

Notable contributions using the multi-grid approach include articles by Ghia *et al.* (1982), Vanka (1986) and Hortmann *et al*. (1990). Ghia *et* a/.(1982) used the vorticity stream function formulation of the Navier-Stokes (NS) equations and employed the strongly implicit technique as their smoothing operator. Very good convergence rates were achieved.

For driven cavity flow, converged solutions were obtained in approximately 20 to 100 equivalent fine grid iterations as the Reynolds number (Re) was varied from 100 to 10000. They found that the multi-grid procedure decreased the computational time by a factor of four over a single-grid calculation.

The article by Vanka (1986) describes a multi grid method based on the primitive variable formulation of the NS equations. Upwind differencing was used, so the solutions are only first order accurate; however, very fine grids are used. Vanka uses the "Symmetrically Coupled Gauss-Seidel" (SCGS) scheme as the basic solver (smoother).The two-dimensional cavity has been treated for Re $<$ 5000. Because it is based on the primitive variable formulation (as opposed to the vorticity-stream function formulation), the method can be easily extended to three dimensions.

Hortmann, Peric and Scheuerer (1990) presented a finite volume multi-grid procedure for the prediction of laminar natural convection flows, enabling efficient and accurate calculations on very fine grids. The method is fully conservative and uses second order central differencing for convection and diffusion fluxes.

Another achievement in the formulation of multigrid methods was the *full multigrid* (FMG) scheme based on the combination of nested iteration techniques and multigrid methods. Multigrid algorithms are now applied to a wide range of problems, primarily to solve linear and nonlinear boundary value problems. Other examples of successful applications are eigenvalue problems, bifurcation problems, parabolic problems, hyperbolic problems, and mixed elliptic/hyperbolic problems, optimization problems, etc.

Most recent developments of solvers based on the multigrid strategy are algebraic multigrid (AMG) methods that resemble the geometric multigrid process utilizing only information contained in the algebraic system to be solved.

In addition to partial differential equations (PDE), Fredholm's integral equations can also be effciently solved by multigrid methods. These schemes can be used to solve reformulated boundary value problems or for the fast solution of *N*-body problems. Another example of applications are lattice field computations and quantum electrodynamics and chromodynamics simulations.

Chapter 2

Computational Fluid Dynamics

2.1 Introduction to CFD

Computational Fluid Dynamics (CFD) is one of the branches of fluid mechanics that uses numerical methods and algorithms to solve and analyze problems that involve fluid flows. Computers are used to perform the millions of calculations required to simulate the interaction of fluids and gases with the complex surfaces used in engineering. However, even with simplified equations and high speed supercomputers, only approximate solutions can be achieved in many cases. More accurate codes that can accurately and quickly simulate even complex scenarios such as supersonic or turbulent flows are an ongoing area of research.

CFD provides a qualitative (and sometimes even quantitative) prediction of fluid flows by means of:

a) mathematical modeling (partial differential equations)
b) numerical methods (discretization and solution techniques)
c) software tools (solvers, pre- and postprocessing utilities)

It also gives an insight into flow patterns that are difficult, expensive or impossible to study using traditional (experimental) techniques.

2.2 Structure of CFD Codes

CFD codes are structured around the numerical algorithms that can be tackle fluid problems. In order to provide easy access to their solving power all commercial CFD packages include sophisticated user interfaces input problem parameters and to examine the results. Hence all codes contain three main elements:

1) Pre-processing:

Preprocessor consist of input of a flow problem by means of an operator –friendly interface and subsequent transformation of this input into form of suitable for the use by the solver.

2) Solver:

Solver consists of discretization by substitution of the approximation into the governing flow equations and subsequent mathematical manipulation and solution of the algebraic equations

3) Post –processing.

These include domain geometry & grid display, vector plots, line & shaded contour plots, 2D and 3D surface plots, Particle tracking, view manipulation (translation, rotation, scaling etc.)

2.3 CFD analysis process

1. Problem statement	information about the flow
2. Mathematical model	IBVP = PDE + IC + BC
3. Mesh generation	nodes/cells, time instants
4. Space discretization	coupled ODE/DAE systems
5. Time discretization	algebraic system Ax = b
6. Iterative solver	discrete function values
7. CFD software	implementation, debugging
8. Simulation run	parameters, stopping criteria
9. Postprocessing	visualization, analysis of data
10.Verification	model validation / adjustment

2.4 Advantages of CFD

1. It provides the flexibility to change design parameters without the expense of hardware changes. It therefore costs less than laboratory or field experiments, allowing engineers to try more alternative designs than would be feasible otherwise.
2. It has a faster turnaround time than experiments.
3. It guides the engineer to the root of problems, and is therefore well suited for troubleshooting.
4. It provides comprehensive information about a flow field, especially in regions where measurements are either difficult or impossible to obtain.

2.5 Governing Equations

Continuity Equation: Conservation of mass

For a given control volume

$$\frac{dm}{dt} = \sum_{in} \dot{m} - \sum_{out} \dot{m}$$

The rate increase of mass within the control volume is equal to the net rate at which mass enters the control volume.

The partial differential form of the continuity equation in cartesian coordinates is

Where $$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y} + w\frac{\partial \rho}{\partial z}$$

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y} + w\frac{\partial \rho}{\partial z}$$

Navier Stokes Equation: Conservation of momentum

$$\sum F_x = m \cdot a_x$$

$$\sum F_y = m \cdot a_y$$

Body forces: Gravity, Centrifugal, Coriolis, Electromagnetic

Surface forces: Normal stress, Tangential stress

Combining forces yields

$$\rho \frac{Du}{Dt} = \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + \sum F_x$$

For an incompressible, constant viscosity flow, the viscous terms simplify significantly (more applicable to gasses than fluids)

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{1}{\rho}\frac{\partial P}{\partial x}+\upsilon\frac{\partial^2 u}{\partial x^2}+\upsilon\frac{\partial^2 u}{\partial y^2}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{1}{\rho}\frac{\partial P}{\partial y}+\upsilon\frac{\partial^2 v}{\partial x^2}+\upsilon\frac{\partial^2 v}{\partial y^2}$$

These set of equations are called the Navier Stokes equation for incompressible flow.

Energy Equation : Conservation of energy

The principle of conservation energy amounts to an application of the first law of thermodynamics to a fluid element as it flows. When applying the first law of thermodynamics to a flowing fluid the instantaneous energy of the fluid is considered to be the sum of the internal energy per unit mass and the kinetic energy per unit mass.

$$\rho\frac{\partial e}{\partial t}+\rho u_k\frac{\partial e}{\partial x_k}=-p\frac{\partial u_k}{\partial x_k}+\frac{\partial}{\partial x_j}\left(k\frac{\partial T}{\partial x_j}\right)+\lambda\left(\frac{\partial u_k}{\partial x_k}\right)^2+\mu\left(\frac{\partial u_j}{\partial x_i}+\frac{\partial u_i}{\partial x_j}\right)\frac{\partial u_j}{\partial x_j}$$

2.6 Discretizaton

Conversion of the governing equations into a system of algebraic equations

Two popular discretization techniques in CFD are

1) Finite difference method

In finite difference method, derivatives in the partial differential equation are approximated by linear combinations of function values at the grid points.

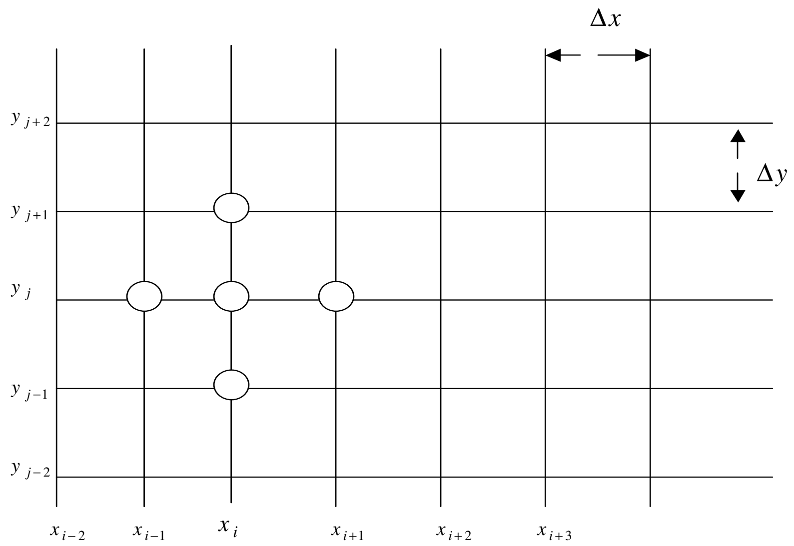
First order derivatives

- Fordward difference $$\frac{\partial u}{\partial x} = \frac{u_{i,j+1} - u_{i,j}}{\Delta x} + O(\Delta x)$$
- Backward difference $$\frac{\partial u}{\partial x} = \frac{u_{i,j} - u_{i,j-1}}{\Delta x} + O(\Delta x)$$
- Central difference $$\frac{\partial u}{\partial x} = \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta x} + O(\Delta x^2)$$

Second order derivative

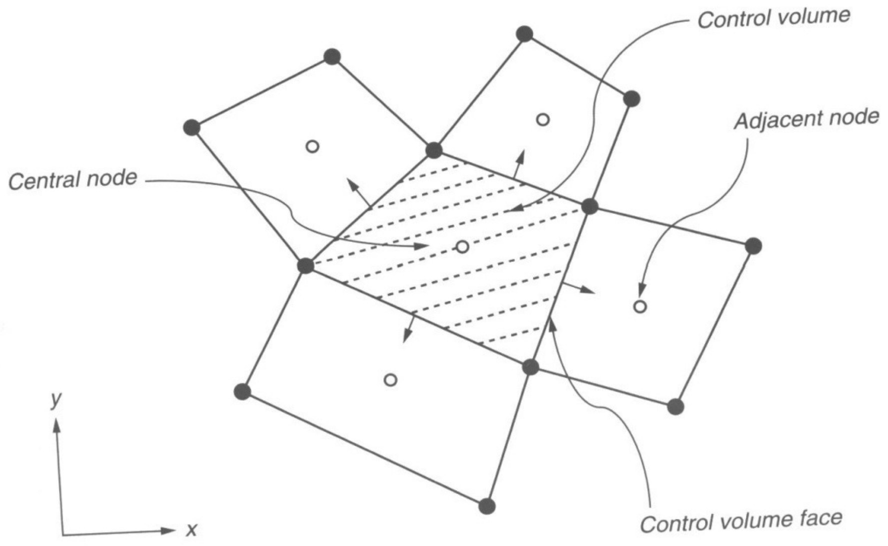
- Central difference $$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + O((\Delta x)^2)$$

For time derivatives $$\frac{\partial u}{\partial t} = \frac{u^{n+1}{}_{i,j} - u^{n}{}_{i,j}}{\Delta t} + O(\Delta t)$$



2) <u>Finite Volume Method</u>

This is the "classical" or standard approach used most often in commercial software and research codes. The governing equations are solved on discrete control volumes. FVM recasts the PDE's (Partial Differential Equations) of the N-S equation in the conservative form and then discretize this equation. This guarantees the conservation of fluxes through a particular control volume. Though the overall solution will be conservative in nature there is no guarantee that it is the actual solution. Moreover this method is sensitive to distorted elements which can prevent convergence if such elements are in critical flow regions.



$$\frac{\partial}{\partial t}\iiint Q\,dV + \iint F\,d\mathbf{A} = 0,$$

Where Q is the vector of conserved variables, F is the vector of fluxes, V is the cell volume, and $\mathbf{A}$ is the cell surface area.

2.7 Stream Function - Vorticity Approach

Stream function ψ for a two dimensional flow is defined such that the flow velocity can be expressed as:

$$\mathbf{u} = \nabla \times \boldsymbol{\psi}$$

where $\boldsymbol{\psi} = (0, 0, \psi)$ if the velocity vector $\mathbf{u} = (u, v, 0)$

In Cartesian coordinate system this is equivalent to

$$u = \frac{\partial \psi}{\partial y}, \qquad v = -\frac{\partial \psi}{\partial x}$$

where u and v are the velocities in the x and y coordinate directions, respectively.

This formulation of the stream function satisfies the two dimensional continuity equation:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

Consider two dimensional plane flow within a Cartesian coordinate system. Continuity states that if we consider incompressible flow into an elemental square, the flow into that small element must equal the flow out of that element.

The total flow into the element is given by:

$$\delta\psi_{in} = u\delta y + v\delta x.$$

The total flow out of the element is given by:

$$\delta\psi_{out} = \left(u + \frac{\partial u}{\partial x}\delta x\right)\delta y + \left(v + \frac{\partial v}{\partial y}\delta y\right)\delta x.$$

Thus we have:

$$\delta\psi_{in} = \delta\psi_{out}$$

$$u\delta y + v\delta x \ = \left(u + \frac{\partial u}{\partial x}\delta x\right)\delta y + \left(v + \frac{\partial v}{\partial y}\delta y\right)\delta x$$

and simplifying to:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0.$$

Substituting the expressions of the stream function into this equation, we have:

$$\frac{\partial^2\psi}{\partial x\partial y} - \frac{\partial^2\psi}{\partial y\partial x} = 0.$$

Vorticity is a concept used in fluid dynamics. In the simplest sense, vorticity is the tendency for elements of the fluid to "spin."

Mathematically, vorticity is a vector field and is defined as the curl of the velocity field:

$$\vec{\zeta} = \vec{\nabla} \times \vec{v}.$$

In Cartesian coordinates, the stream function can be found from vorticity using the following Poisson's equation:

$$\nabla^2\psi = -\omega$$

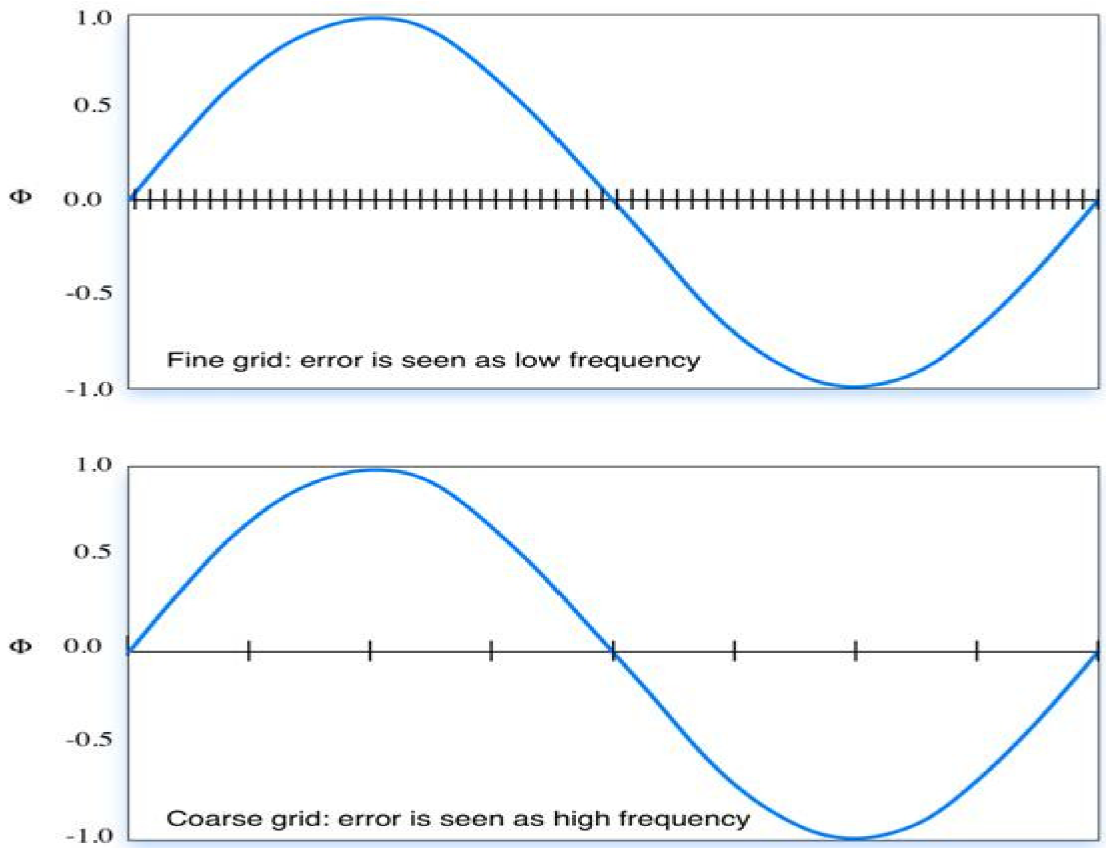
where $\vec{\omega} = (0, 0, \omega)$ and $\vec{\omega} = \vec{\nabla} \times \vec{v}.$

Chapter 3

Multigrid Algorithm

3.1 Why Use Multigrid?

The convergence rate of standard iterative solvers (Gauss-Siedel, Jacobi, SOR) has a tendency to 'stall', i.e. to fail in effectively reducing errors after a few number of iterations. The problem is more prominent when the meshes are refined. In fact, standard solvers behave much better on coarse grids. A close inspection of this behavior reveals that the convergence rate is a function of the error field frequency, i.e. the gradient of the error from node to node. If the error is distributed in a high frequency mode, the convergence rate is fast. However, after the first few iterations, the error field is smoothed out (low frequency) and the convergence rate deteriorates.



3.2 Multigrid Theory

When we solve fluid dynamics, heat transfer or other engineering problems numerically by using iterative procedures, one of the major issues that concern us is getting a converged solution on a fine grid. The studies on convergence history of our solution procedure reveal that in the initial stages of our iterative procedure there is a rapid reduction in residual and as we march iteratively, the efficiency of smoothing the error falls down and eventually the solution procedure may stall. Here we will discuss the factors that affect the convergence of conventional solution procedures on single grid and then study how we can overcome these problems by using multi-grid algorithms.

Given a set of finite difference equations

$A^k\ u^k = f^k$

for a general elliptic equation, any iterative procedure such as Gauss-Seidel, Jacobi, incomplete LU factorization, etc., is known to converge rapidly for the first few iterations and very slowly thereafter. Fourier analysis of the error reduction process shows that these conventional iterative procedures are most efficient in smoothing out the errors of wave lengths comparable to the mesh size, but are inefficient in removing low frequency components. However, the low frequency components on any grid are relatively larger on grids that are coarser than the grid in question. Multi-grid methods seek to exploit the high frequency smoothing of relaxation methods.

The multi-grid technique is based on the premise that each frequency range of error must be smoothed on the grid where it is most suitable to do so. Consequently, the multi-grid technique cycles between coarser and finer grids until all the frequency components of error are appropriately smoothed. The multigrid concept is distinct from the philosophy of starting a fine grid solution from an interpolated coarse grid converged solution.

In the latter concept, only a better starting guess is provided. Therefore the starting residual is smaller than a raw guess, but the asymptotic rate of convergence is not improved. The multi-grid method, on the other hand, cycles between hierarchies of computational grids, so that error components of all frequency ranges are efficiently removed.

3.3 Basic Algorithm:

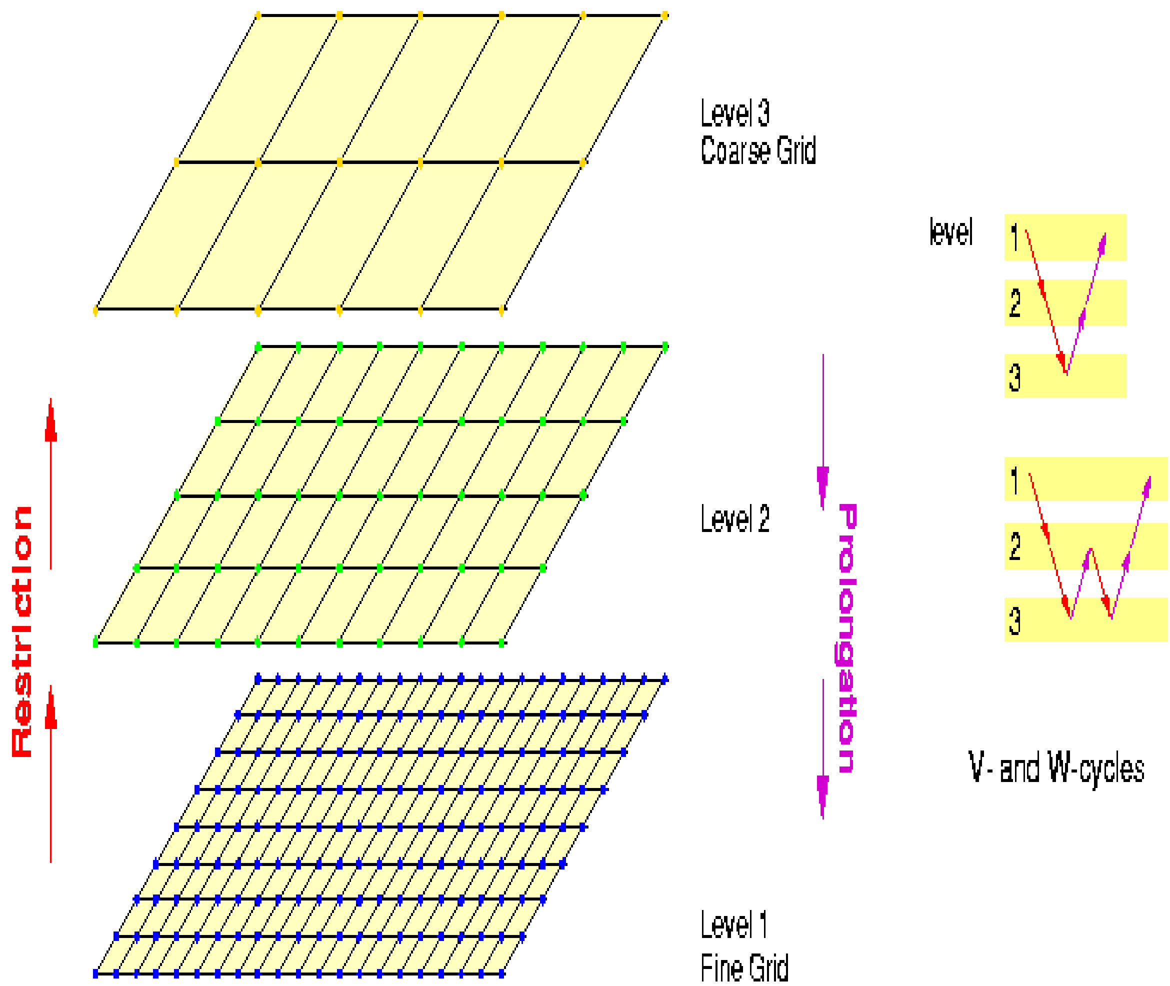
Replace problem on fine grid by an approximation on a coarser grid.

Solve the coarse grid problem approximately, and use the solution as a starting guess for the fine-grid problem, which is then iteratively updated.

Solve the coarse grid problem recursively, i.e. by using a still coarser grid approximation, etc.

3.4 Two Dimensional Multigrid System

Multigrid methods are a state-of-the art technique to solve large systems of linear equations A x = b, where $A \in \mathbb{R}^{n \times n}$ and $x, b \in \mathbb{R}^n$. This system can be represented as a graph of n nodes where an edge (i,j) represents a non-zero coefficient. To simplify the following illustration, we assume that graph to be a regular two dimensional grid. The basic idea of multigrid is to define a hierarchy of grids as illustrated in the figure. Each node at the coarser grid level represents a set of nodes at the finer level. Coefficients at some grid level i are derived from coefficients at grid level i+1 (prolongation) or from coefficients at grid level i-1 (restriction). The grid hierarchy is traversed in V or W-cycles. On each level of the hierarchy an iterative solver is called.



3.5 Numerical Formulation

The residual equation: Let ν be the approximation to the solution of Au=f, where the residual r=f-Av. Then the error e=u-v satisfies Ae=r.

After relaxing on Au=f on the fine grid, e will be smooth so the coarse grid can approximate e well. This will be cheaper and e should be more oscillatory there, so relaxation will be more effective.

Therefore, we go to a coarse grid and relax on the residual equation Ae=r.

3.6 Multigrid Algorithm

Pre smoothing : Relax on Au=f on Ω^h to get approximation ν^h

Computation of residual : Compute $r = f - A\nu^h$

Restriction of residual : $r_{2h} = I_h^{2h} r_h$

Calculation of error : Relax on Ae=r on Ω^{2h} to obtain an approximation to the error, e^{2h}

Correct the approximation $v^h \rightarrow v^h + I_{2h}^h e^{2h}$

Post smoothing : Relax on Au=f on Ω_h.

3.7 Restriction and Prolongation

Restriction is used to transfer the value of the residual from a grid Ω_n to a grid Ω_{n-1}; relaxation is then a *fine-to-coarse* process.

In a two-dimensional scheme is convenient to use two different restriction operators, namely the *full-weighting* and the *half-weighting*. In the simple case of a homogeneous, square grid, one has:

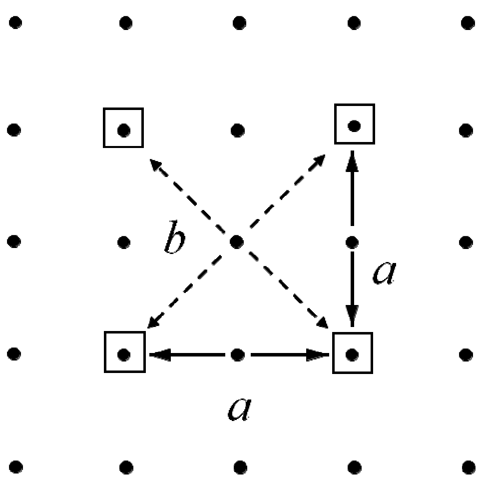
$$\begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix} \quad \text{Full Weighting} \qquad \begin{bmatrix} 0 & \frac{1}{8} & 0 \\ \frac{1}{8} & \frac{1}{2} & \frac{1}{8} \\ 0 & \frac{1}{8} & 0 \end{bmatrix} \quad \text{Half Weighting}$$

Prolongation is used to transfer the computed error from a grid Ω_{n-1} to a grid Ω_n. It is a coarse-to-fine process and, in two dimensions, can be described as follows (see figure below):

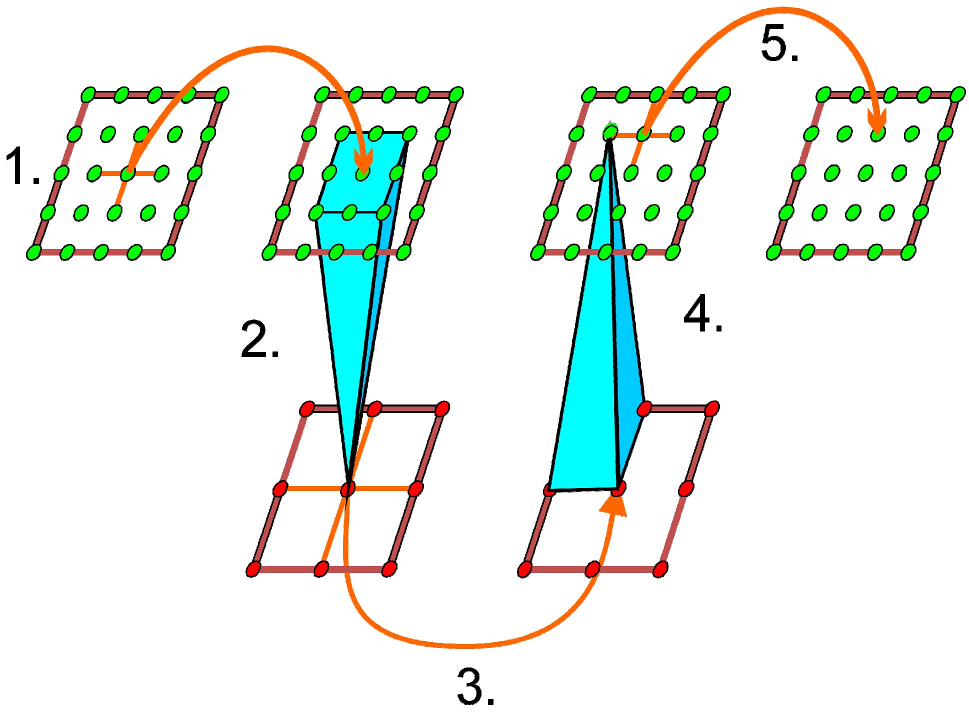
Values on points on the fine grid which correspond to points on the coarse one (framed points) are just copied.

Values on points of type *a* are linearly interpolated from the two closer values on the coarse grid.

Values on points of type *b* are bilinearly interpolated from the four closer values on the coarse grid.



3.8 Two Grid Iteration to solve Av= f on grid Ω_n



1) Pre-smoothing on Ω_n : Smooth $\mathbf{v}^i$ on the grid Ω_n using some suitable relaxation scheme.

2) Restriction of **r** to Ω_{n-1}: Compute the residual according to $\mathbf{r}^i=\mathbf{f}-A\mathbf{v}^i$ and transfer it to the coarser grid Ω_{n-1}.

3) Solution of A**e** = **r** on Ω_{n-1}: Solve exactly the residual equation $A\mathbf{e}^i=\mathbf{r}^i$ on grid Ω_{n-1}. The exact value of the error $\mathbf{e}^i$ is then known on Ω_{n-1}.

4) Prolongation of **e** to Ω_n : Interpolate $\mathbf{e}^i$ to the finer grid W_n, calculate $\mathbf{v}^{i+1}$ using $\mathbf{v}^{i+1}=\mathbf{v}^i-\mathbf{e}^i$.

5) Post-smoothing on Ω_n: Smooth $\mathbf{v}^{i+1}$ on the grid W_n by applying some relaxation method.

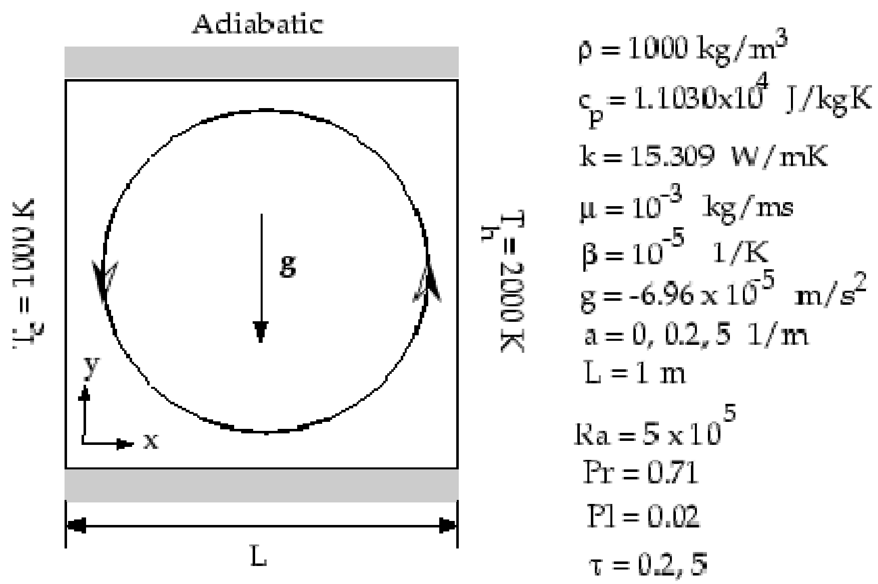
Chapter 4

Application and Results

4.1 Problem Statement

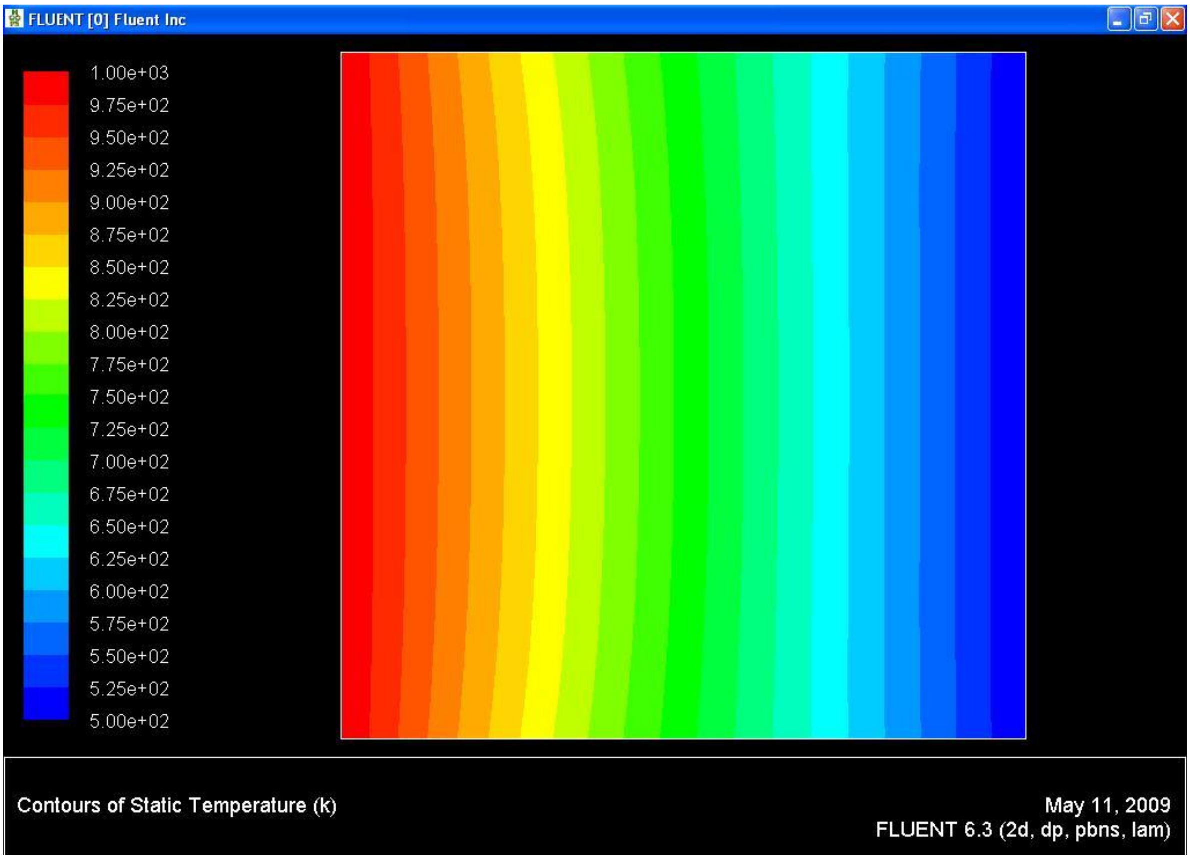
Natural Convection in a Square Enclosure

A Square box of side L has a hot right wall at 2000K, a cold left wall at 1000K, and adiabatic top and bottom walls. Gravity acts downwards. A buoyant force develops because of thermally induced density gradients. The objective is to compute the flow and temperature patterns in the box, as well as the wall heat flux. The working fluid has a Prandlt number of approximate 0.71, and the Reyleigh number based L is 5e+05. This means the flow is inherently laminar. The Boussinesq assumption is to used to model buoyancy.

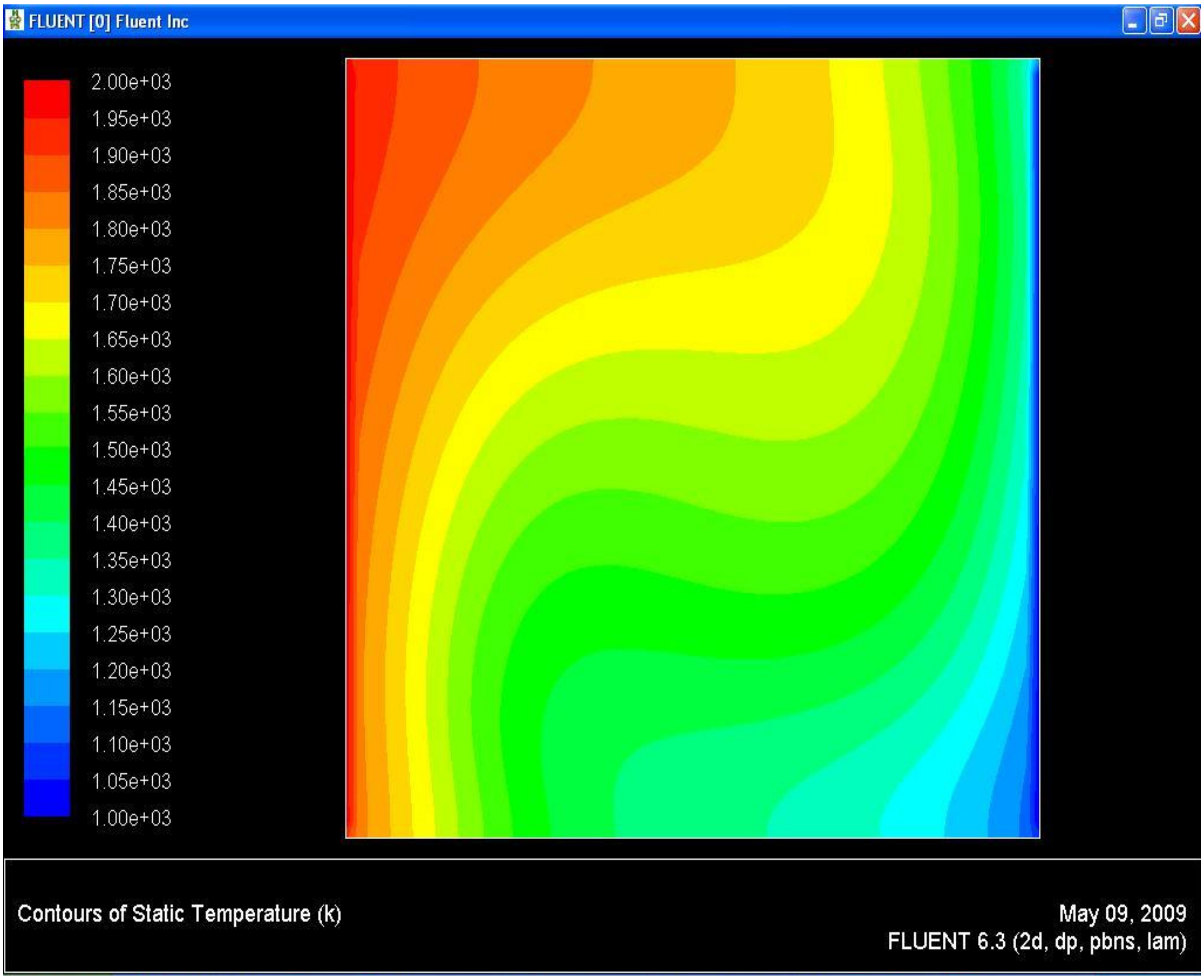


4.2 Results

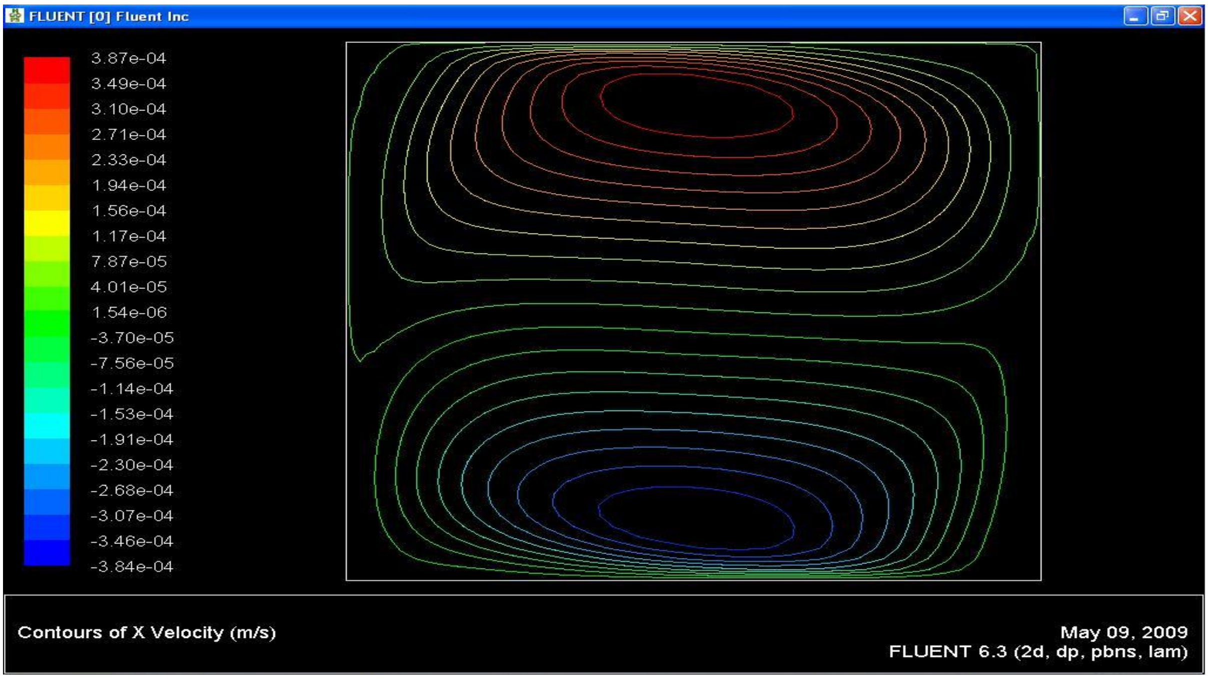
The results consists of CFD diagrams which pictures the contours of static temperature in case of pure conduction and convection and contours of X velocity, Y velocity and stream function in case of convection.



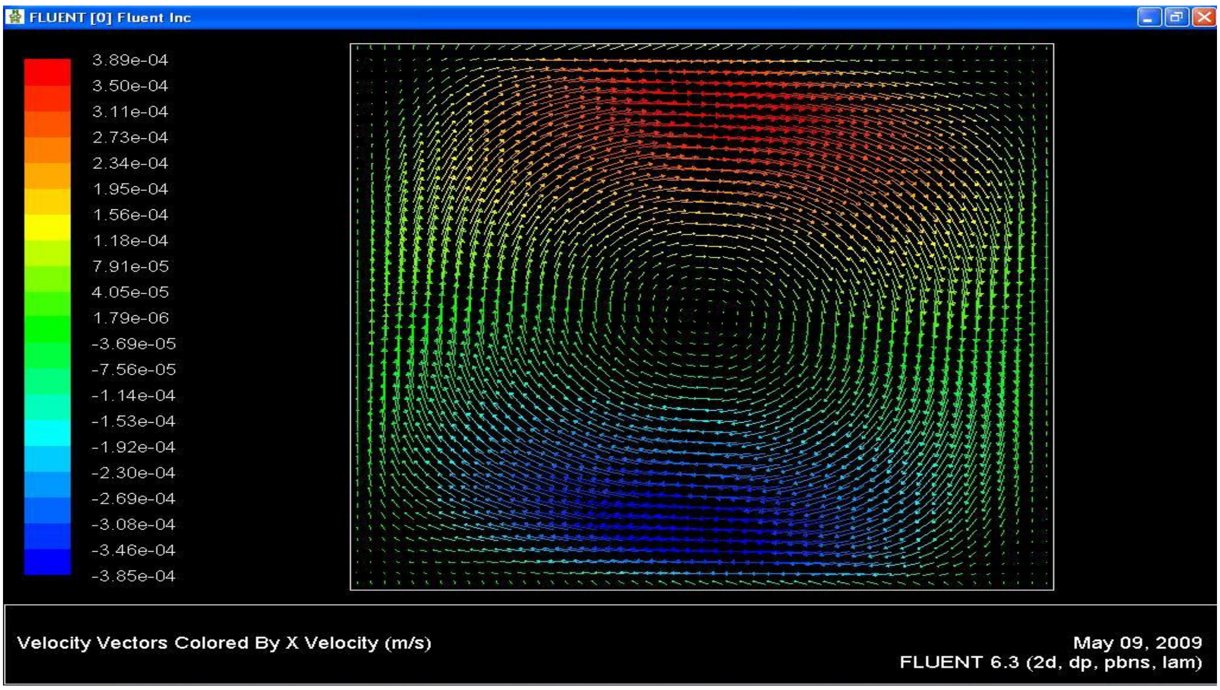
Contours of Static Temperature(K) in case of Pure Conduction



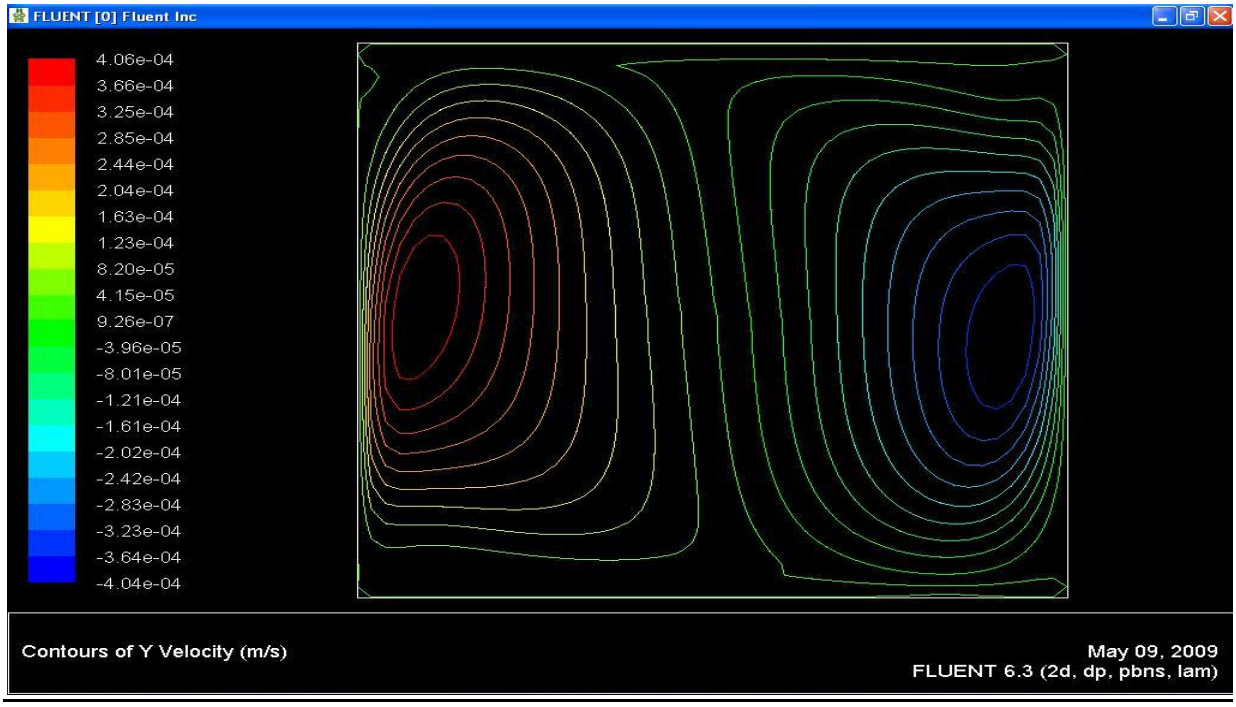
Contours of Static Temperature(K) in case of Pure Convection



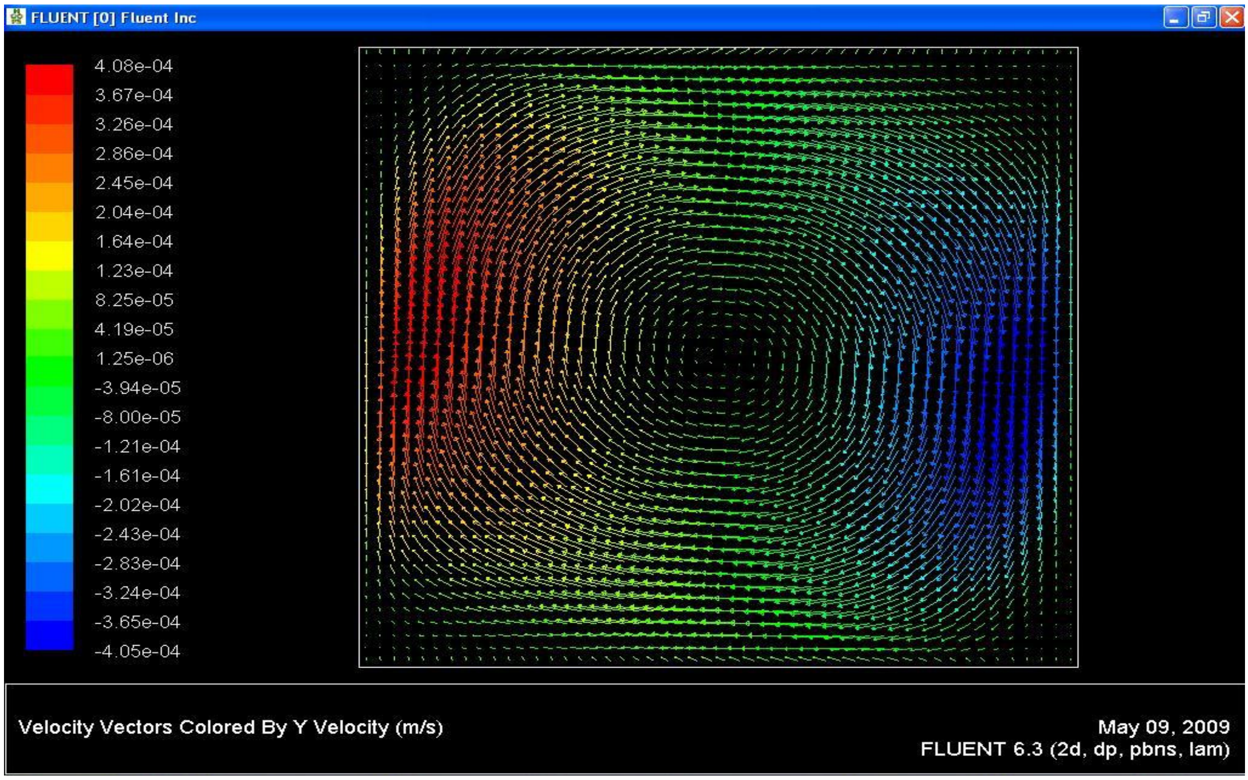
Contours of X velocity (m/s)



Velocity Vectors Coloured By X Velocity(m/s)



Contours of Y velocity(m/s)



Velocity Vectors Colored By Y Velocity(m/s)

Chapter 5

Conclusion

The CFD results obtained using Fluent software were in perfect agreement with the analytical results. Thus they are verified. Also, the time required to solve the model problem was less than that taken using the simple grid technique. Multigrid algorithm uses the best of both coarse and fine grids and thus a fast and accurate result is obtained.

Multigrid techniques have been applied in diverse fields like weather prediction, solid mechanics, tomography (CAT scan), image segmentation, quantum chemistry and VLSI design. But multigrid algorithm suffers from certain drawbacks like currently they are several orders of magnitude slower for non-elliptic steady-state problems. Also it takes careful tuning to get the algorithm to work on a new problem.

References

[1] Nader Ben Cheikh, *Natural Convection in a tall enclosure using multigrid method. Numerical Methods for Partial Differential Equations*, Academic Press, New York, Jan 2007

[2] J.Steelant, *Multigrid methods for compressible Navier Stokes equation in low speed flows, Journal of Computational and Apllied Mathematics* (1997)

[3] Claudio Cianfrini, *Natural convection in tilted square cavities with differentially heated opposite walls.*

[4] Ruben S Montero, *Robust multigrid algorithm for the Navier Stokes Equations, Journal of Computational Physics,173*.

[5] A. Brandt, S. McCormick and J. Ruge, *Multigrid Methods for Differential Eigenproblems. SIAM J. Sci. Stat. Comput.* 4 (1983), 244-260.

[6] A. Brandt, *Multi-grid techniques: 1984 guide with applications to fluid dynamics*. GMD-Studien, St. Augustin, Germany,1984.